Project Two: Logistic Regression and Random Forests

For Project Two, you have been asked to create different models analyzing a Heart Disease data set. Before beginning work on the project, be sure to read through the Project Two Guidelines and Rubric to understand what you need to do and how you will be graded on this assignment. Be sure to carefully review the Project Two Summary Report template, which contains all of the questions that you will need to answer about the regression analyses you are performing.

For this project, you will be writing all the scripts yourself. You may reference the textbook and your previous work on the problem sets to help you write the scripts.

Scenario

You are a data analyst researching risk factors for heart disease at a university hospital. You have access to a large set of historical data that you can use to analyze patterns between different health indicators (e.g. fasting blood sugar, maximum heart rate, etc.) and the presence of heart disease. You have been asked to create different logistic regression models that predict whether or not a person is at risk for heart disease. A model like this could eventually be used to evaluate medical records and look for risks that might not be obvious to human doctors. You have also been asked to create a classification random forest model to predict the risk of heart disease and a regression random forest model to predict the maximum heart rate achieved.

There are several variables in this data set, but you will be working with the following important variables:

Variable	What does it represent?
age	The person's age in years
sex	The person's sex (1 = male, 0 = female)
cp	The type of chest pain experienced (0=no pain, 1=typical angina, 2=atypical angina, 3=non-anginal pain)
trestbps	The person's resting blood pressure
chol	The person's cholesterol measurement in mg/dl
fbs	The person's fasting blood sugar is greater than 120 mg/dl (1 = true, 0 = false)
restecg	Resting electrocardiographic measurement (0=normal, 1=having ST-T wave abnormality, 2=showing probable or definite left ventricular hypertrophy by Estes' criteria)
thalach	The person's maximum heart rate achieved
exang	Exercise-induced angina (1=yes, 0=no)
oldpeak	ST depression induced by exercise relative to rest ('ST' relates to positions on the ECG plot)
slope	The slope of the peak exercise ST segment (1=upsloping, 2=flat, 3=downsloping)
ca	The number of major vessels (0-3)
target	Heart disease (0=no, 1=yes)

Install Libraries

In the following code block, you will install appropriate libraries to use in this project.

Click the **Run** button on the toolbar to run this code.

Note: The code section below will first install three R packages: "ResourceSelection", "pROC" and "rpart.plot". Please do not move to the next step until the packages are fully installed. This will take some time. Once the installation is complete, this step will print first 6 rows of the data set.

In [1]:
```
print("This step will first install three R packages. Please wait until the pa
ckages are fully installed.")
print("Once the installation is complete, this step will print 'Installation c
omplete!'")

install.packages("ResourceSelection")
install.packages("pROC")
install.packages("rpart.plot")

print("Installation complete!")
```

```
[1] "This step will first install three R packages. Please wait until the pac
kages are fully installed."
[1] "Once the installation is complete, this step will print 'Installation co
mplete!'"

Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)
Installing package into '/home/codio/R/x86_64-pc-linux-gnu-library/3.4'
(as 'lib' is unspecified)

[1] "Installation complete!"
```

Prepare Your Data Set

In the following code block, you have been given the R code to prepare your data set.

Click the **Run** button on the toolbar to run this code.

In [2]:
```
heart_data <- read.csv(file="heart_disease.csv", header=TRUE, sep=",")

# Converting appropriate variables to factors
heart_data <- within(heart_data, {
   target <- factor(target)
   sex <- factor(sex)
   cp <- factor(cp)
   fbs <- factor(fbs)
   restecg <- factor(restecg)
   exang <- factor(exang)
   slope <- factor(slope)
   ca <- factor(ca)
   thal <- factor(thal)
})

head(heart_data, 10)

print("Number of variables")
ncol(heart_data)

print("Number of rows")
nrow(heart_data)
```

A data.frame: 10 × 14

age	sex	cp	trestbps	chol	fbs	restecg	thalach	exang	oldpeak	slope	ca	th
<int>	<fct>	<fct>	<int>	<int>	<fct>	<fct>	<int>	<fct>	<dbl>	<fct>	<fct>	<
62	1	2	130	231	0	1	146	0	1.8	1	3	3
58	0	0	130	197	0	1	131	0	0.6	1	0	2
60	0	3	150	240	0	1	171	0	0.9	2	0	2
63	1	0	140	187	0	0	144	1	4.0	2	2	3
62	1	0	120	267	0	1	99	1	1.8	1	2	3
63	0	2	135	252	0	0	172	0	0.0	2	0	2
43	1	0	150	247	0	1	171	0	1.5	2	0	2
42	1	2	120	240	1	1	194	0	0.8	0	0	3
59	1	2	126	218	1	1	134	0	2.2	1	1	1
48	1	0	124	274	0	0	166	0	0.5	1	0	3

```
[1] "Number of variables"
```

14

```
[1] "Number of rows"
```

303

Model #1 - First Logistic Regression Model

You have been asked to create a logistic regression model for heart disease *(target)* using the variables age *(age)*, resting blood pressure *(trestbps)*, exercised induced angina (exang) and maximum heart rate achieved *(thalach)*. Before writing any code, review Section 3 of the Summary Report template to see the questions you will be answering about your logistic regression model.

Run your scripts to get the outputs of your regression analysis. Then use the outputs to answer the questions in your summary report.

Note: Use the + (plus) button to add new code blocks, if needed.

In [3]:
```
# First Logistic Regression Model
model1 <- glm(target ~ age + trestbps + exang + thalach , data = heart_data, f
amily = "binomial")

summary(model1)
```

```
Call:
glm(formula = target ~ age + trestbps + exang + thalach, family = "binomial",
    data = heart_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.0935  -0.7944   0.4954   0.8133   2.2343

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.021121   1.784194  -0.572   0.5671
age         -0.017549   0.017144  -1.024   0.3060
trestbps    -0.014888   0.008337  -1.786   0.0741 .
exang1      -1.624981   0.305774  -5.314 1.07e-07 ***
thalach      0.031095   0.007275   4.274 1.92e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 417.64  on 302  degrees of freedom
Residual deviance: 323.14  on 298  degrees of freedom
AIC: 333.14

Number of Fisher Scoring iterations: 4
```

In [4]:
```
#Perform Hosmer-Lemeshow goodness of fit test
library(ResourceSelection)


print("Hosmer-Lemeshow Goodness of Fit Test Model 1")
hl = hoslem.test(model1$y, fitted(model1), g=50)
hl
```

```
ResourceSelection 0.3-6          2023-06-27

[1] "Hosmer-Lemeshow Goodness of Fit Test Model 1"

        Hosmer and Lemeshow goodness of fit (GOF) test

data:  model1$y, fitted(model1)
X-squared = 44.622, df = 48, p-value = 0.612
```

In [5]:
```
#Confusion Matrix
# Predict heart disease risk or no heart disease risk for the data set
heart_model_data <- heart_data[c('age', 'trestbps', 'exang', 'thalach')]
pred <- predict(model1, newdata=heart_model_data, type='response')

# If the predicted probability of heart disease is >=0.50 then predict heart d
isease (yes='1'), otherwise predict no heart
# disease (no='0')
depvar_pred = as.factor(ifelse(pred >= 0.5, '1', '0'))

# This creates the confusion matrix
conf.matrix <- table(heart_data$target, depvar_pred)[c('0','1'),c('0','1')]
rownames(conf.matrix) <- paste("Actual", rownames(conf.matrix), sep = ": heart
disease=")
colnames(conf.matrix) <- paste("Prediction", colnames(conf.matrix), sep = ": h
eart disease=")

# Print nicely formatted confusion matrix
print("Confusion Matrix Model 1")
format(conf.matrix,justify="centre",digit=2)
```

```
[1] "Confusion Matrix Model 1"
```

A matrix: 2 × 2 of type chr

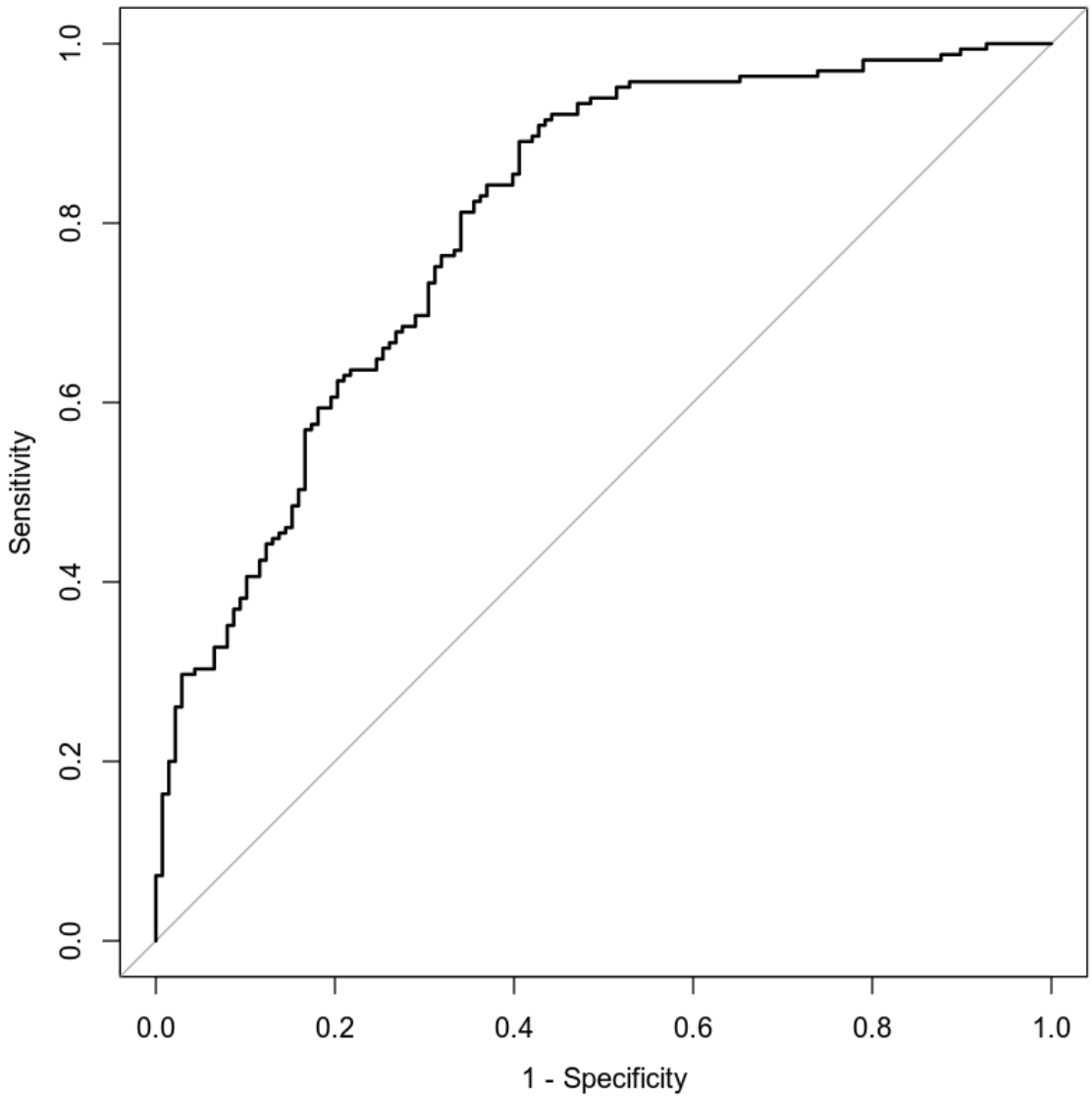
	Prediction: heart disease=0	Prediction: heart disease=1
Actual: heart disease=0	89	49
Actual: heart disease=1	31	134

In [6]:
```
#Receiver Operating Characteristic (ROC) Curve
library(pROC)

labels <- heart_data$target
predictions <- model1$fitted.values

roc <- roc(labels ~ predictions)

print("Area Under the Curve (AUC)")
round(auc(roc),4)

print("ROC Curve")
# True Positive Rate (Sensitivity) and False Positive Rate (1 - Specificity)
plot(roc, legacy.axes = TRUE)
```

```
Type 'citation("pROC")' for a citation.

Attaching package: 'pROC'

The following objects are masked from 'package:stats':

    cov, smooth, var

Setting levels: control = 0, case = 1
Setting direction: controls < cases

[1] "Area Under the Curve (AUC)"
```

0.8007

```
[1] "ROC Curve"
```



In [7]:
```
#Making predictions using Model 1
print("Prediction: 50 years old (age=50), Resting Blood Pressure of 122 (trest
bps=122), has exercised induced angina (exang='1'), max. heart rate of 140 (th
alach=140)")
newdata1 <- data.frame(age=50, trestbps=122, exang='1', thalach=140)
pred1 <- predict(model1, newdata1, type='response')
round(pred1, 4)

print("Prediction: 50 years old (age=50), Resting Blood Pressure of 130 (trest
bps=130), no exercised induced angina (exang='0'), max. heart rate of 165 (tha
lach=165)")
newdata2 <- data.frame(age=50, trestbps=130, exang='0', thalach=165)
pred2 <- predict(model1, newdata2, type='response')
round(pred2, 4)
```

```
[1] "Prediction: 50 years old (age=50), Resting Blood Pressure of 122 (trestb
ps=122), has exercised induced angina (exang='1'), max. heart rate of 140 (th
alach=140)"
```

1: 0.2716

```
[1] "Prediction: 50 years old (age=50), Resting Blood Pressure of 130 (trestb
ps=130), no exercised induced angina (exang='0'), max. heart rate of 165 (tha
lach=165)"
```

1: 0.7853

Model #2 - Second Logistic Regression Model

You have been asked to create a logistic regression model for heart disease *(target)* using the variables age of the individual *(age)*, resting blood pressure (trestbps), type of chest pain *(cp)* and maximum heart rate achieved *(thalach)*. You also have to include the quadratic term for age and the interaction term between age and maximum heart rate achieved. Before writing any code, review Section 4 of the Summary Report template to see the questions you will be answering about your model.

Run your scripts to get the outputs of your analysis. Then use the outputs to answer the questions in your summary report.

Note: Use the + (plus) button to add new code blocks, if needed.

In [8]:
```
# First Logistic Regression Model
model2 <- glm(target ~ age + trestbps + cp + thalach + I(age^2) + age:thalach
, data = heart_data, family = "binomial")

summary(model2)
```

```
Call:
glm(formula = target ~ age + trestbps + cp + thalach + I(age^2) +
    age:thalach, family = "binomial", data = heart_data)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.6961  -0.7537   0.2925   0.7123   2.3058

Coefficients:
              Estimate Std. Error z value Pr(>|z|)
(Intercept) -1.556e+01  1.054e+01  -1.476  0.13988
age          1.744e-01  2.669e-01   0.653  0.51357
trestbps    -1.958e-02  8.978e-03  -2.181  0.02916 *
cp1          1.913e+00  4.437e-01   4.313 1.61e-05 ***
cp2          2.037e+00  3.473e-01   5.867 4.45e-09 ***
cp3          1.777e+00  5.477e-01   3.245  0.00117 **
thalach      1.363e-01  5.119e-02   2.663  0.00775 **
I(age^2)     8.424e-04  1.750e-03   0.481  0.63025
age:thalach -1.867e-03  8.909e-04  -2.095  0.03616 *
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 417.64  on 302  degrees of freedom
Residual deviance: 293.67  on 294  degrees of freedom
AIC: 311.67

Number of Fisher Scoring iterations: 5
```

In [9]:
```
#Perform Hosmer-Lemeshow goodness of fit test
library(ResourceSelection)


print("Hosmer-Lemeshow Goodness of Fit Test Model 2")
h2 = hoslem.test(model2$y, fitted(model2), g=50)
h2
```

```
[1] "Hosmer-Lemeshow Goodness of Fit Test Model 2"

        Hosmer and Lemeshow goodness of fit (GOF) test

data:  model2$y, fitted(model2)
X-squared = 52, df = 48, p-value = 0.3209
```

In [10]:
```
#Confusion Matrix
# Predict heart disease risk or no heart disease risk for the data set
heart_model_data2 <- heart_data[c('age', 'trestbps', 'cp', 'thalach')]
pred2 <- predict(model2, newdata=heart_model_data2, type='response')

# If the predicted probability of heart disease is >=0.50 then predict heart d
isease (yes='1'), otherwise predict no heart
# disease (no='0')
depvar_pred2 = as.factor(ifelse(pred2 >= 0.5, '1', '0'))

# This creates the confusion matrix
conf.matrix2 <- table(heart_data$target, depvar_pred2)[c('0','1'),c('0','1')]
rownames(conf.matrix2) <- paste("Actual", rownames(conf.matrix2), sep = ": hea
rt disease=")
colnames(conf.matrix2) <- paste("Prediction", colnames(conf.matrix2), sep = ":
heart disease=")

# Print nicely formatted confusion matrix
print("Confusion Matrix Model 2")
format(conf.matrix2,justify="centre",digit=2)
```

```
[1] "Confusion Matrix Model 2"
```

A matrix: 2 × 2 of type chr

	Prediction: heart disease=0	Prediction: heart disease=1
Actual: heart disease=0	102	36
Actual: heart disease=1	36	129

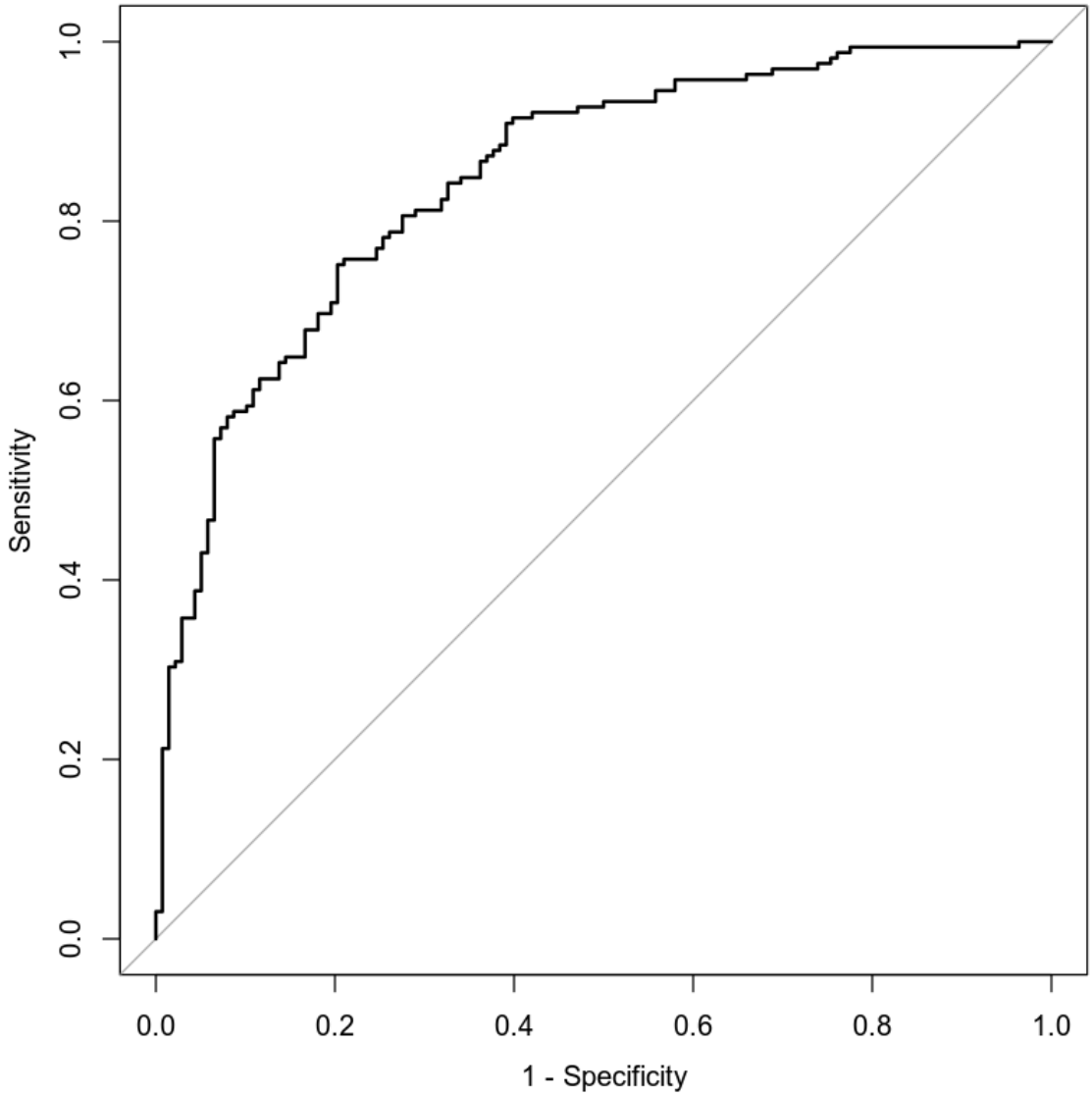
In [11]:
```
#Receiver Operating Characteristic (ROC) Curve
library(pROC)

labels <- heart_data$target
predictions2 <- model2$fitted.values

roc2 <- roc(labels ~ predictions2)

print("Area Under the Curve (AUC)")
round(auc(roc2),4)

print("ROC Curve")
# True Positive Rate (Sensitivity) and False Positive Rate (1 - Specificity)
plot(roc2, legacy.axes = TRUE)
```

```
Setting levels: control = 0, case = 1
Setting direction: controls < cases
```

```
[1] "Area Under the Curve (AUC)"
```

0.8478

```
[1] "ROC Curve"
```



In [12]:
```
#Making predictions using Model 2
print("Prediction: 50 years old (age=50), Resting Blood Pressure of 115 (trest
bps=115), no chest pain (cp='0'), max. heart rate of 133 (thalach=133)")
newdata3 <- data.frame(age=50, trestbps=115, cp='0', thalach=133)
pred3 <- predict(model2, newdata3, type='response')
round(pred3, 4)

print("Prediction: 50 years old (age=50), Resting Blood Pressure of 125 (trest
bps=125), typical angina(cp='1'), max. heart rate of 155 (thalach=155)")
newdata4 <- data.frame(age=50, trestbps=125, cp='1', thalach=155)
pred4 <- predict(model2, newdata4, type='response')
round(pred4, 4)
```

```
[1] "Prediction: 50 years old (age=50), Resting Blood Pressure of 115 (trestb
ps=115), no chest pain (cp='0'), max. heart rate of 133 (thalach=133)"
```

1: 0.2188

```
[1] "Prediction: 50 years old (age=50), Resting Blood Pressure of 125 (trestb
ps=125), typical angina(cp='1'), max. heart rate of 155 (thalach=155)"
```

1: 0.8007

Random Forest Classification Model

You have been asked to create a random forest classification model for the presence of heart disease *(target)* using the variables age *(age)*, sex *(sex)*, chest pain type *(cp)*, resting blood pressure *(trestbps)*, cholesterol measurement *(chol)*, resting electrocardiographic measurement *(restecg)*, exercise-induced angina *(exang)*, and number of major vessels *(ca)*. Before writing any code, review Section 5 of the Summary Report template to see the questions you will be answering about your model.

Run your scripts to get the outputs of your regression analysis. Then use the outputs to answer the questions in your summary report.

Note: Use the + (plus) button to add new code blocks, if needed.

In [13]:
```
set.seed(6522048)

# Partition the data set into training and testing data
samp.size = floor(0.85*nrow(heart_data))

# Training set
print("Number of rows for the training set")
train_ind = sample(seq_len(nrow(heart_data)), size = samp.size)
train.data1 = heart_data[train_ind,]
nrow(train.data1)

# Testing set
print("Number of rows for the validation set")
test.data1 = heart_data[-train_ind,]
nrow(test.data1)
```

```
[1] "Number of rows for the training set"
```

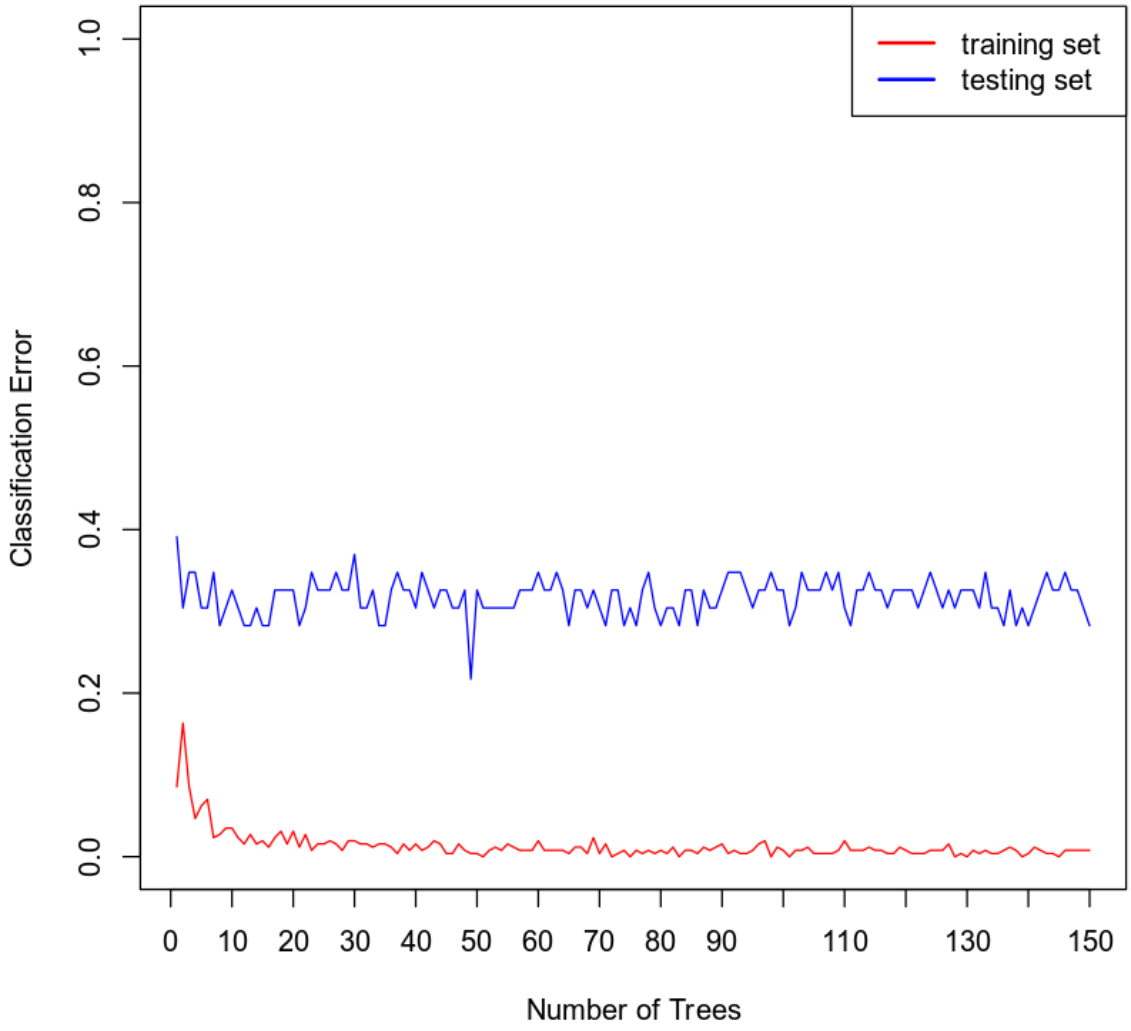
257

```
[1] "Number of rows for the validation set"
```

46

In [14]:
```
set.seed(6522048)
library(randomForest)

# checking
#=====================================================================
train = c()
test = c()
trees = c()

for(i in seq(from=1, to=150, by=1)) {
    #print(i)

    trees <- c(trees, i)

    model_rf1 <- randomForest(target ~ age + sex + cp + trestbps + chol + rest
ecg + exang + ca, data=train.data1, ntree = i)

    train.data.predict <- predict(model_rf1, train.data1, type = "class")
    conf.matrix1 <- table(train.data1$target, train.data.predict)
    train_error = 1-(sum(diag(conf.matrix1)))/sum(conf.matrix1)
    train <- c(train, train_error)

    test.data.predict <- predict(model_rf1, test.data1, type = "class")
    conf.matrix2 <- table(test.data1$target, test.data.predict)
    test_error = 1-(sum(diag(conf.matrix2)))/sum(conf.matrix2)
    test <- c(test, test_error)
}

#matplot (trees, cbind (train, test), ylim=c(0,0.5) , type = c("l", "l"), lwd=
2, col=c("red","blue"), ylab="Error", xlab="number of trees")
#legend('topright',legend = c('training set','testing set'), col = c("red","bl
ue"), lwd = 2 )

plot(trees, train,type = "l",ylim=c(0,1.0),xaxp = c(0,150,15), col = "red", xl
ab = "Number of Trees", ylab = "Classification Error")
lines(test, type = "l", col = "blue")
legend('topright',legend = c('training set','testing set'), col = c("red","blu
e"), lwd = 2 )
```

```
randomForest 4.6-14
Type rfNews() to see new features/changes/bug fixes.
```



In [15]:
```
set.seed(6522048)
library(randomForest)
model_rf2 <- randomForest(target ~ age + sex + cp + trestbps + chol + restecg
+ exang + ca, data=train.data1, ntree = 25)

# Confusion matrix
print("============================================================================
=============================================")
print('Confusion Matrix: TRAINING set based on random forest model built using
25 trees')
train.data.predict <- predict(model_rf2, train.data1, type = "class")

# Construct the confusion matrix
conf.matrix3 <- table(train.data1$target, train.data.predict)[,c('0','1')]
rownames(conf.matrix3) <- paste("Actual", rownames(conf.matrix3), sep = ": hea
rt disease=")
colnames(conf.matrix3) <- paste("Prediction", colnames(conf.matrix3), sep = ":
heart disease=")

# Print nicely formatted confusion matrix
format(conf.matrix3,justify="centre",digit=2)


print("============================================================================
=============================================")
print('Confusion Matrix: TESTING set based on random forest model built using
25 trees')
test.data.predict <- predict(model_rf2, test.data1, type = "class")

# Construct the confusion matrix
conf.matrix4 <- table(test.data1$target, test.data.predict)[,c('0','1')]
rownames(conf.matrix4) <- paste("Actual", rownames(conf.matrix4), sep = ": hea
rt disease=")
colnames(conf.matrix4) <- paste("Prediction", colnames(conf.matrix4), sep = ":
heart disease=")

# Print nicely formatted confusion matrix
format(conf.matrix4,justify="centre",digit=2)
```

```
[1] "============================================================================
=============================================="
[1] "Confusion Matrix: TRAINING set based on random forest model built using
25 trees"
```

A matrix: 2 × 2 of type chr

	Prediction: heart disease=0	Prediction: heart disease=1
Actual: heart disease=0	119	1
Actual: heart disease=1	1	136

```
[1] "============================================================================
=============================================="
[1] "Confusion Matrix: TESTING set based on random forest model built using 2
5 trees"
```

A matrix: 2 × 2 of type chr

	Prediction: heart disease=0	Prediction: heart disease=1
Actual: heart disease=0	11	7
Actual: heart disease=1	9	19

Random Forest Regression Model

You have been asked to create a random forest regression model for maximum heart rate achieved using the variables age *(age)*, sex *(sex)*, chest pain type *(cp)*, resting blood pressure *(trestbps)*, cholesterol measurement *(chol)*, resting electrocardiographic measurement *(restecg)*, exercise-induced angina *(exang)*, and number of major vessels *(ca)*. Before writing any code, review Section 6 of the Summary Report template to see the questions you will be answering about your model.

Run your scripts to get the outputs of your analysis. Then use the outputs to answer the questions in your summary report.

Note: Use the + (plus) button to add new code blocks, if needed.

In [16]:
```
set.seed(6522048)

# Partition the data set into training and testing data
samp.size = floor(0.80*nrow(heart_data))

# Training set
print("Number of rows for the training set")
train_ind1 = sample(seq_len(nrow(heart_data)), size = samp.size)
train.data2 = heart_data[train_ind1,]
nrow(train.data2)

# Testing set
print("Number of rows for the validation set")
test.data2 = heart_data[-train_ind1,]
nrow(test.data2)
```

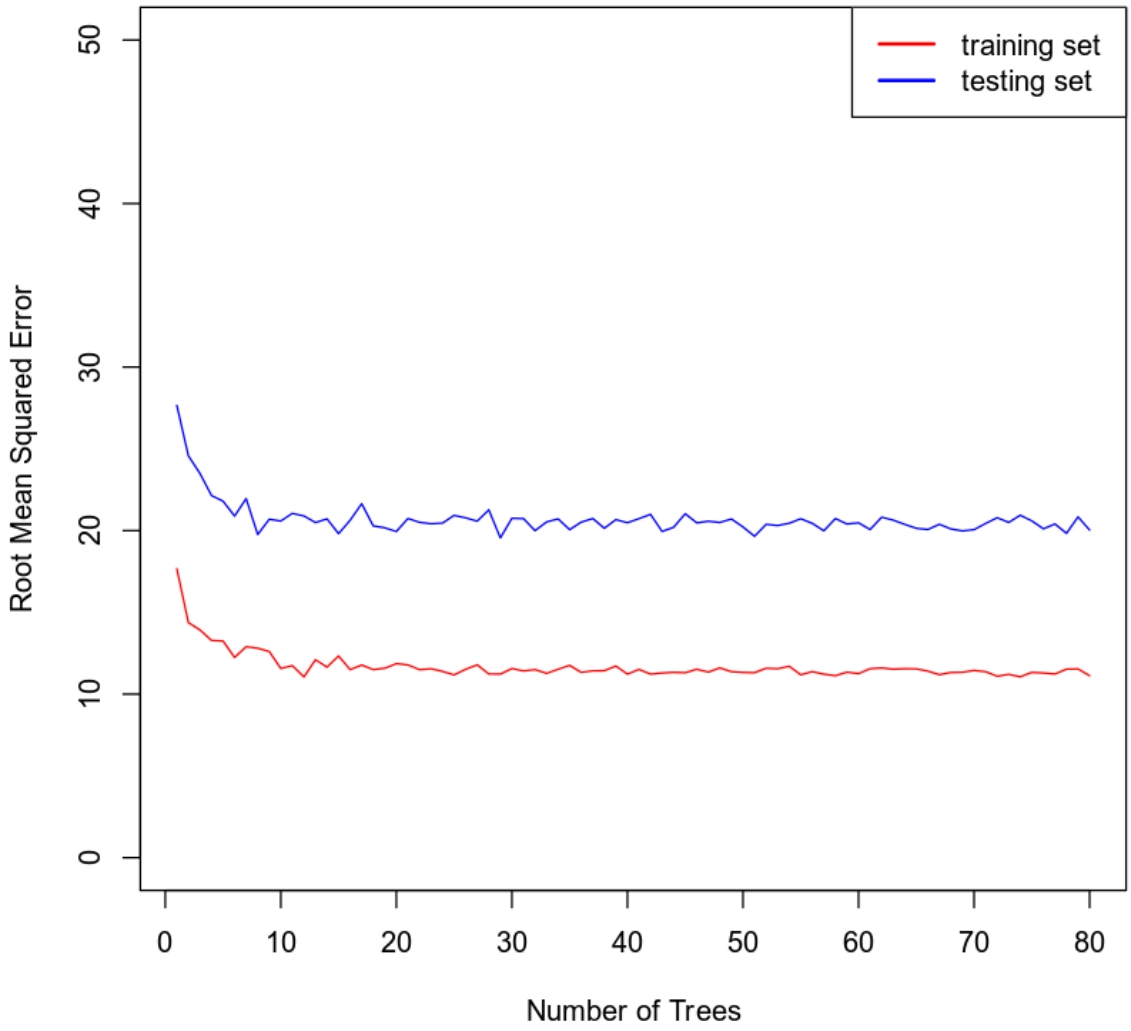
```
[1] "Number of rows for the training set"
```

```
242
```

```
[1] "Number of rows for the validation set"
```

```
61
```

In [17]:
```
set.seed(6522048)
library(randomForest)

# Root mean squared error
RMSE = function(pred, obs) {
    return(sqrt( sum( (pred - obs)^2 )/length(pred) ) )
}


# checking
#=====================================================================
train = c()
test = c()
trees = c()

for(i in seq(from=1, to=80, by=1)) {
    trees <- c(trees, i)
    model_rf3 <- randomForest(thalach ~ age + sex + cp + trestbps + chol + res
tecg + exang + ca, data=train.data2, ntree = i)

    pred <- predict(model_rf3, newdata=train.data2, type='response')
    rmse_train <-  RMSE(pred, train.data2$thalach)
    train <- c(train, rmse_train)

    pred <- predict(model_rf3, newdata=test.data2, type='response')
     rmse_test <-  RMSE(pred, test.data2$thalach)
    test <- c(test, rmse_test)
}

plot(trees, train,type = "l",ylim=c(0,50),xaxp = c(0,80,8), col = "red", xlab
= "Number of Trees", ylab = "Root Mean Squared Error")
lines(test, type = "l", col = "blue")
legend('topright',legend = c('training set','testing set'), col = c("red","blu
e"), lwd = 2 )
```


training set
testing set
Root Mean Squared Error
0
10
20
30
40
50
0
10
20
30
40
50
60
70
80
Number of Trees

In [18]:
```
set.seed(6522048)
library(randomForest)
model_rf4 <- randomForest(thalach ~ age + sex + cp + trestbps + chol + restecg
+ exang + ca, data=train.data2, ntree = 20)


# Root mean squared error
RMSE = function(pred, obs) {
    return(sqrt( sum( (pred - obs)^2 )/length(pred) ) )
}

print("================================================================================
================================================")
print('Root Mean Squared Error: TRAINING set based on random forest model buil
t using 20 trees')
pred <- predict(model_rf4, newdata=train.data2, type='response')
RMSE(pred, train.data2$thalach)


print("================================================================================
================================================")
print('Root Mean Squared Error: TESTING set based on random forest model built
using 20 trees')
pred <- predict(model_rf4, newdata=test.data2, type='response')
RMSE(pred, test.data2$thalach)
```

```
[1] "================================================================================
================================================"
[1] "Root Mean Squared Error: TRAINING set based on random forest model built
using 20 trees"
```

11.6281588266415

```
[1] "================================================================================
================================================"
[1] "Root Mean Squared Error: TESTING set based on random forest model built
using 20 trees"
```

21.1139392413712

End of Project Two Jupyter Notebook

The HTML output can be downloaded by clicking **File**, then **Download as**, then **HTML**. Be sure to answer all of the questions in the Summary Report template for Project Two, and to include your completed Jupyter Notebook scripts as part of your submission.